

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A computer-readable medium having computer-executable instructions for performing steps for processing Input/Output ("I/O") requests, comprising:
 - receiving an I/O request from an application thread;
 - performing an I/O operation in response to the I/O request; and
 - upon completion of the I/O operation, determining whether to boost a priority of the application thread according to criteria based on a status of I/O operations performed for the application thread, future I/O operations to be performed for the application thread or whether a period of time since a last time the priority of the application thread was boosted has reached a threshold length.
2. (Original) A computer-readable medium as in claim 1, having further computer-executable instructions for performing steps of:
 - if the step of determining determines not to boost the priority of the application thread, performing a further I/O operation for the application thread, and determining again whether to boost the priority of the application thread.
3. (Original) A computer-readable medium as in claim 1, wherein the application thread posts a data buffer in connection with the I/O request, and the step of performing the I/O operation includes copying data into the I/O buffer.
4. (Original) A computer-readable medium as in claim 1, having further computer-executable instructions for performing the step of boosting the priority of the application thread when the step of determining determines that the priority of the application is to be boosted.
5. (Original) A computer-readable medium as in claim 4, wherein the step of boosting boosts the priority of the application thread by a pre-selected level.

6. (Original) A computer-readable medium as in claim 5, wherein the pre-selected level is fixed.

7. (Currently Amended) A computer-readable medium as in claim 1, wherein the criteria for determining whether to boost the priority of the application thread includes an analysis of the number of ~~whether there are more~~ I/O operations to be performed in the future for the application thread.

8. (Original) A computer-readable medium as in claim 1, wherein the criteria for determining whether to boost the priority of the application thread include whether a number of I/O operations performed in a current thread context for the application thread has reached a threshold number.

9. (Currently Amended) A computer-readable medium as in claim 7~~4~~, wherein the criteria for determining whether to boost the priority of the application thread include ~~whether a period of time since a last time the priority of the application thread was boosted has reached a threshold length~~ determining the number of I/O operations to be performed in the future and if the number is below a threshold, refraining from increasing the priority of the application thread until the I/O operations are complete.

10. (Currently Amended) A method of processing Input/Output ("I/O") requests, comprising:

receiving an I/O request from an application thread;
performing an I/O operation in response to the I/O request; and
upon completion of the I/O operation, determining whether to boost a priority of the application thread according to criteria based on ~~a status of I/O operations performed for the application thread~~ future I/O operations to be performed for the application thread or whether a period of time since a last time the priority of the application thread was boosted has reached a threshold length

11. (Original) A method as in claim 10, having further steps of:
if the step of determining determines not to boost the priority of the application thread, performing a further I/O operation for the application thread, and determining again whether to boost the priority of the application thread.
12. (Original) A method as in claim 10, wherein the application thread posts a data buffer in connection with the I/O request, and the step of performing the I/O operation includes copying data into the I/O buffer.
13. (Original) A method as in claim 10, having a further step of boosting the priority of the application thread when the step of determining determines that the priority of the application is to be boosted.
14. (Original) A method as in claim 13, wherein the step of boosting boosts the priority of the application thread by a pre-selected level.
15. (Original) A method as in claim 14, wherein the pre-selected level is fixed.
16. (Currently Amended) A method as in claim 10, wherein the criteria for determining whether to boost the priority of the application thread include includes an analysis of the number of whether there are more I/O operations to be performed in the future for the application thread.
17. (Original) A method as in claim 10, wherein the criteria for determining whether to boost the priority of the application thread include whether a number of I/O operations performed in a current thread context has reached a threshold number.
18. (Currently Amended) A method as in ~~claim 1~~claim 15, wherein the criteria for determining whether to boost the priority of the application thread include ~~whether a period of time since a last time the priority of the application thread was boosted has reached a threshold length~~determining the number of I/O operations to be performed in the future and if the number is below a threshold, refraining from increasing the priority of the application thread until the I/O operations are complete.

19. (Currently Amended) A computer system comprising:
an application thread making an I/O request;
a system thread for responding to the I/O request, the system thread being
programmed to receive the I/O request from the application thread, perform an I/O operation
in response to the I/O request, and upon completion of the I/O operation, determine whether
to boost a priority of the application thread according to criteria based on ~~a status of I/O~~
~~operations performed for the application thread, future I/O operations to be performed for the~~
application thread or whether a period of time since a last time the priority of the application
thread was boosted has reached a threshold length.

20. (Original) A computer system as in claim 19, wherein the system thread is
further programmed to perform steps of:
if the system thread determines not to boost the priority of the application thread,
performing a further I/O operation for the application thread, and determining again whether
to boost the priority of the application thread.

21. (Original) A computer system as in claim 19, wherein the application thread
posts a data buffer in connection with the I/O request, and the I/O operation performed by the
system thread includes copying data into the I/O buffer.

22. (Original) A computer system as in claim 19, wherein the system thread is
programmed to boost the priority of the application thread by a pre-selected level when the
system thread determines that the priority of the application is to be boosted.

23. (Original) A computer system as in claim 22, wherein the pre-selected level is
fixed.

24. (Currently Amended) A computer system as in claim 19, wherein the criteria
for determining whether to boost the priority of the application thread include includes an
analysis of the number of ~~whether there are more~~ I/O operations to be performed in the
future for the application thread.

25. (Original) A computer system as in claim 19, wherein the criteria for determining whether to boost the priority of the application thread include whether a number of I/O operations performed in a current thread context for the application thread has reached a threshold number.

26. (Currently Amended) A computer system as in claim 23, wherein the criteria for determining whether to boost the priority of the application thread include ~~whether a period of time since a last time the priority of the application thread was boosted has reached a threshold length~~ determining the number of I/O operations to be performed in the future and if the number is below a threshold, refraining from increasing the priority of the application thread until the I/O operations are complete.